

# Tarea 3

## Redes Sociales y Económicas

Daniel Ramos, Sergi Fornés & Jordi Vanrell

1. Implementar el cálculo de las matrices de los núcleos de Neumann. La función debería tener dos parámetros: la matriz de adyacencia y el grado de difusión.

```
NK <- function(A, gamma) {  
  G <- graph_from_adjacency_matrix(A)  
  # Calculamos el máximo grado de difusión posible  
  MIdg <- max(degree(G,mode="in"))  
  MOdg <- max(degree(G,mode="out"))  
  max_gamma <- 1/(min(MIdg,MOdg))  
  # Comprobamos que gamma no es excesivamente elevado  
  if(gamma > max_gamma) {  
    return(paste("Grado de difusión demasiado alto. El valor debe estar entre 0 y",  
                 round(max_gamma,3)))  
  }  
  K_ <- t(A) %*% A  
  T_ <- A %*% t(A)  
  sumK <- 0  
  sumT <- 0  
  for (i in 0:20) {  
    sumK <- sumK + (gamma^i)*(matrix.power(K_,i))  
    sumT <- sumT + (gamma^i)*(matrix.power(T_,i))  
  }  
  K_g <- K_ %*% sumK  
  T_g <- T_ %*% sumT  
  output = list()  
  output$K_g <- K_g  
  output$T_g <- T_g  
  return(output)  
}
```

- 
2. Aplicar el algoritmo a la matriz de adyacencia de páginas webs: `webPageLinks.txt`.

```
data <- as.matrix(read.table("webPageLinks.txt"))  
gf <- graph_from_adjacency_matrix(data)
```

Con estos datos, responda las siguientes preguntas:

- Describe how the HITS algorithm relates to the Neumann Kernel technique.

El algoritmo HITS puntúa los nodos en función de su indegree y outdegree. Si un nodo tiene indegree elevado querrá decir que muchos otros nodos le apuntan, y por lo tanto será un nodo vecino común a muchos nodos. Por lo tanto, este nodo será próximo en términos de importancia a muchos nodos en la matriz K de Neumann Kernel.

- How does the decay factor  $\gamma$  affect the output of the Neumann Kernel algorithm?

El grado de difusión es el parámetro que da importancia a los vecinos lejanos. Un grado de difusión pequeño produce outputs cercanos a “Shared Nearest Neighbor”, ya que este algoritmo solo tiene en cuenta los vecinos más cercanos. Por el otro lado, un grado de difusión elevado produce outputs cercanos a “HITS”, puesto que tendrá en cuenta todos los vecinos comunes, incluso los más lejanos.

```
# Calculamos el máximo grado de difusión posible
MIdg <- max(degree(gf,mode="in"))
MOdg <- max(degree(gf,mode="out"))
max_gamma <- 1/(min(MIdg,MOdg))

# Calculamos todos los scores que nos interesan
hub <- hub_score(gf, weights = NA)$vector
auth <- authority_score(gf, weights = NA)$vector
NeuKern_maxK <- NK(data,max_gamma)$K_g %>% colSums
NeuKern_maxT <- NK(data,max_gamma)$T_g %>% colSums
NeuKern_minK <- NK(data,0)$K_g %>% colSums
NeuKern_minT <- NK(data,0)$T_g %>% colSums

# Calculamos las correlaciones
cor(hub,NeuKern_maxT)
```

```
## [1] 1
```

```
cor(hub,NeuKern_minT)
```

```
## [1] 0.978092
```

```
cor(auth,NeuKern_maxK)
```

```
## [1] 1
```

```
cor(auth,NeuKern_minK)
```

```
## [1] 0.871743
```

Efectivamente, como vemos en la tabla de correlaciones, cuando el  $\gamma$  aumenta, HITS y Neumann Kernel convergen.

Correlación	$\gamma = 0$	$\gamma = \frac{1}{\min(\Delta^+(G), \Delta^-(G))}$
Hub vs Importance ( $\hat{T}_\gamma$ )	0.978	1
Authority vs Relevance ( $\hat{K}_\gamma$ )	0.872	1